

· A guide to Microsoft
languages for the
MS® OS/2, MS-DOS®,
and XENIX®
operating systems.

Precision tools for the art of programming

File Edit View Search Run Debug Calls

argc: 2

argv: 0x3d2a

Add Watch...

Delete Last Watch

Delete All Watch

☒ **Trace On**

☒ **Screen Swapping On**

Toggle Breakpoint

Clear All Breakpoint

main (argc, argv)

int argc;

char **argv;

{

static char *pcklst[MA

int lines;

FILE *infile();

FILE *fp;

int miscount;

/* number of

/* get input file */

fp = infile(argc, argv);

/* load 'er up */

Microsoft®

Microsoft offers the technology you need to perfect your art.

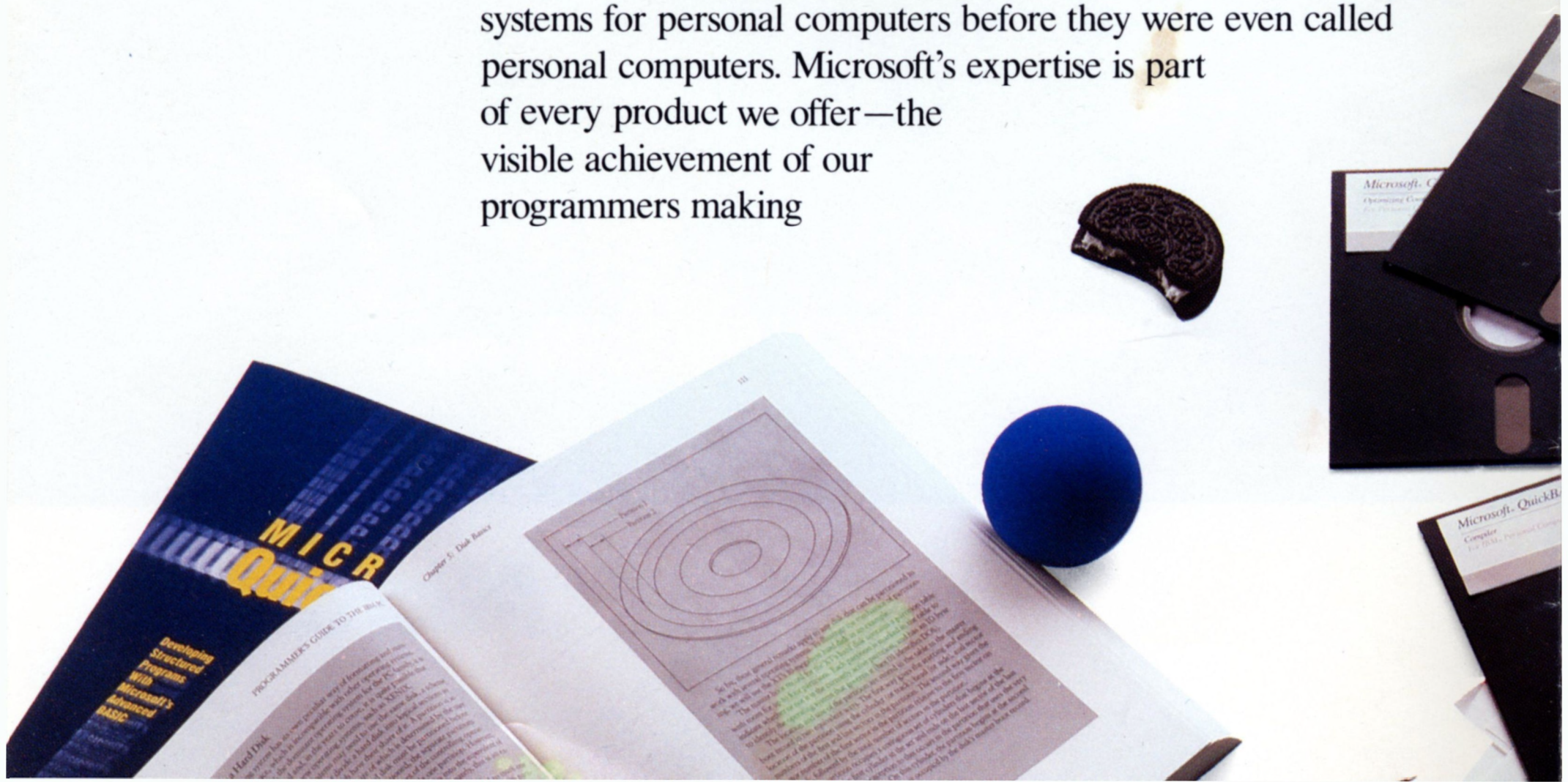
Computer programming can take you into late nights and lost weekends. But there's help at hand to make that extra effort pay off with code that's fast, clean, and bug-free—the Microsoft programming languages.

Microsoft languages for the MS OS/2, MS-DOS, and XENIX operating systems are the precision tools you need to get the job done quickly, accurately, and elegantly. Tools that are powerful microcomputer implementations of leading programming languages. Tools that make those late nights not only easier, but shorter and more rewarding, too.

And Microsoft offers a wider selection of language solutions than anyone else. Our *Quick* line is the perfect introduction to programming. These products supply everything you need in one integrated package and provide capabilities even experienced users will appreciate. For high-end needs, we offer languages that use the latest optimizing technology to create the fastest possible code.

Now that you have at least one Microsoft language product, we'd like to introduce you to the entire MS OS/2, MS-DOS, and XENIX family of languages. There's sure to be one that can handle the programming challenges you encounter. If you'd like more information on specific languages, send in the enclosed order form for free data sheets.

Whichever products you choose, you'll know that they come from the company that began writing languages and operating systems for personal computers before they were even called personal computers. Microsoft's expertise is part of every product we offer—the visible achievement of our programmers making



programming easier for others. And more efficient, too, so you can keep those all-night coding sessions to a minimum.

But not all of Microsoft's support is in this package. Telephone hotlines, timely updates, and state-of-the-art on-line support mean our products will work for you now and in the future. (To make sure you're informed about these services, take a moment right now to fill out the registration card included in this package.) Also available is a selection of Microsoft Press books that cover basic programming techniques and tips on using individual products more efficiently. It all adds up to an unmatched commitment to quality from the leader in personal computer languages.

Microsoft language products for MS OS/2, MS-DOS, and XENIX.

You provide the art of programming.
We'll supply the technology.



Microsoft® QuickBASIC.

Instant programming means instant results!

In the past, if you wanted to program in BASIC you had to choose between the immediacy of using an interpreter and the fast program execution of a compiler. Now, no choice is necessary, because instant programming is here with Microsoft QuickBASIC version 4.0.

Microsoft QuickBASIC version 4.0 is a revolutionary concept

in BASIC programming. There's no waiting to run or debug your program because there's no compile step. You can run your program, stop to edit and debug, then continue running. Whenever you change your code, Microsoft QuickBASIC 4.0 automatically incorporates the change so quickly—usually at 150,000 lines per minute*—that it seems instantaneous!

Microsoft QuickBASIC version 4.0 also makes it easier to write multiple-module programs. The new built-in code outliner

keeps track of all subprograms and functions in an individual module and lets you edit two of them at a time.

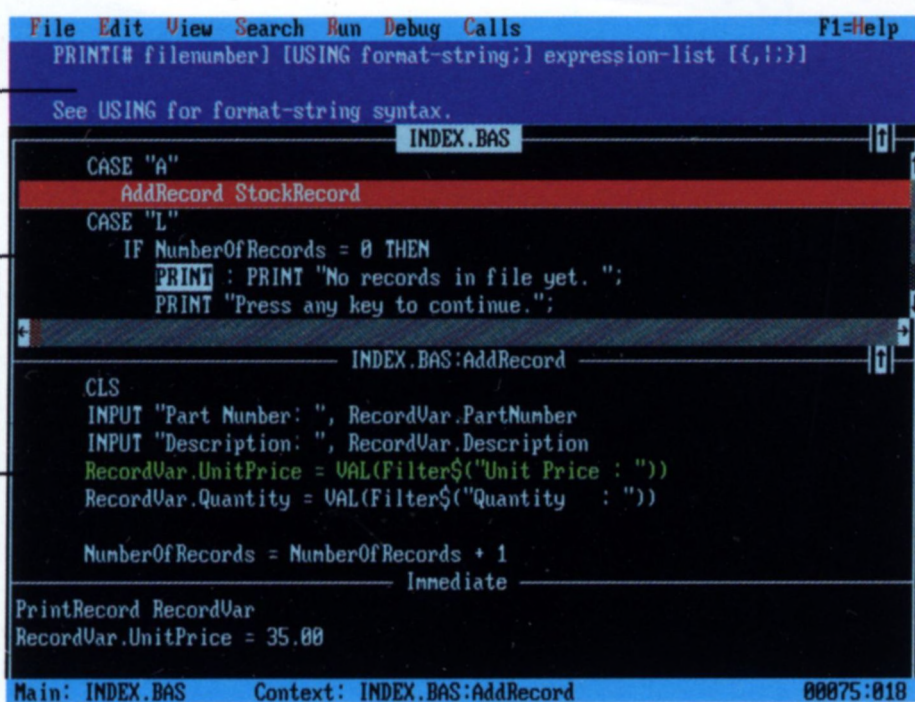
Microsoft QuickBASIC also features:

- Stand-alone, multiple-module executables that can be generated in a single operation
- Full-screen windowing editor that lets you edit two parts of your program simultaneously and supports WordStar®-compatible keystrokes
- Automatic syntax error checking after each line of code is typed
- Enhanced language features, including records, recursion, and data arrays up to available memory
- Automatic support for Intel® 8087 and 80287 math coprocessors
- Support for interlanguage calling to other Microsoft language products, including BASIC Compiler 6.0, C, QuickC™, FORTRAN, Macro Assembler, and Pascal

System requirements

- 320K available user memory
- DOS 2.1 or higher
- One double-sided disk drive

*On an IBM® PC/AT® at 8 MHz.



```
File Edit View Search Run Debug Calls F1=Help
PRINT[# filename] [USING format-string:] expression-list [(,;)]
See USING for format-string syntax.

INDEX.BAS
CASE "A"
  AddRecord StockRecord
CASE "L"
  IF NumberOfRecords = 0 THEN
    PRINT : PRINT "No records in file yet. ";
    PRINT "Press any key to continue.";

INDEX.BAS:AddRecord
CLS
INPUT "Part Number: ", RecordVar.PartNumber
INPUT "Description: ", RecordVar.Description
RecordVar.UnitPrice = VAL(Filter$("Unit Price : "))
RecordVar.Quantity = VAL(Filter$("Quantity : "))

NumberOfRecords = NumberOfRecords + 1
Immediate
PrintRecord RecordVar
RecordVar.UnitPrice = 35.00
Main: INDEX.BAS Context: INDEX.BAS:AddRecord 00075:018
```

- Multiwindow editing and debugging give you a flexible and powerful BASIC programming environment.
- Context-sensitive help gives you quick answers

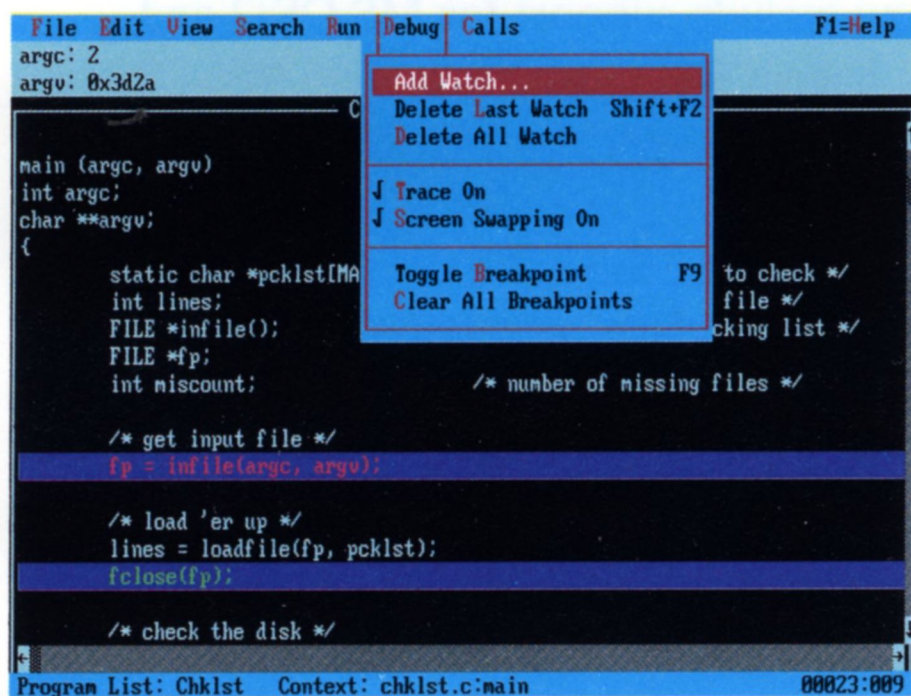
Microsoft QuickC Compiler.

Total integration. Totally amazing!

Even if you've never programmed in C before, you can explore the power of this versatile computer language with Microsoft QuickC Compiler.

Microsoft QuickC Compiler features an easy-to-use debugger that's fully integrated with the editor and compiler, so you have all the tools at your fingertips all the time.

The full-screen editor gives you complete cut-copy-paste convenience, as well as detailed error messages that help you solve problems quickly. The compiler not only crunches your code at 10,000 lines per minute,* but also finds as many as 26 errors at a time. Then it puts you right back into the editor with the cursor on the first error, ready for you to begin making changes. And recompiling is easy, because Microsoft QuickC automatically generates MAKE files for you—just choose the modules you want



Your programming tools are at your command all the time. Editing, compiling, and debugging are quick and easy because there's no switching between tools.

included, and QuickC does the rest.

Microsoft QuickC's source-level debugger is integrated right into the compiler. You can see exactly what your code is doing during execution, making your debugging faster and more efficient. Single-step, animate, or trace through your source code. Set dynamic breakpoints to stop execution at any point you choose. And track local and global variables so you can see values change as the program runs.

Microsoft QuickC Compiler also features:

- Full compatibility with Microsoft C Optimizing Compiler
- Extensive, easy-to-use documentation
- Context-sensitive help that displays information on statement syntax and library routines
- Graphics library
- Two math libraries (8087/80287 support and floating-point emulation)
- Support for the Microsoft Mouse

System requirements

- 448K available user memory
- DOS 2.1 or higher
- Two double-sided disk drives

*On an IBM PC/AT at 8 MHz.

New support for Microsoft Operating System/2.

MS OS/2 is the operating system of the future for personal computers. It's designed to take advantage of the advanced capabilities of the Intel 80286 processor, providing huge memory access and multitasking in protected mode.

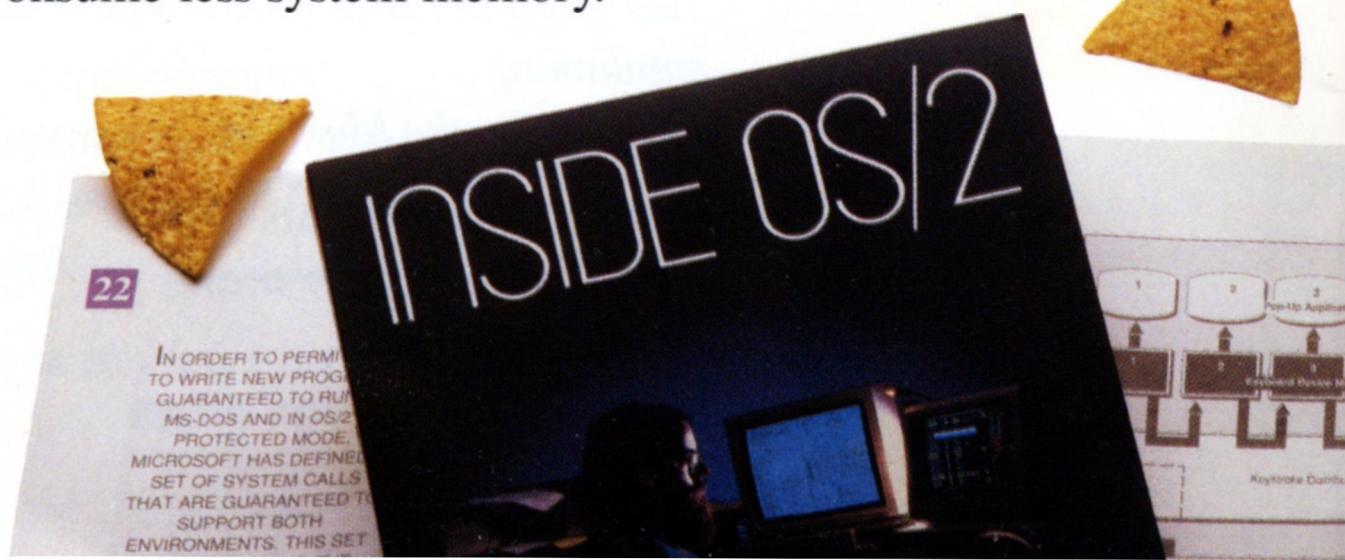
As part of our ongoing commitment to provide you with the best and most advanced programming tools possible, Microsoft languages now include support for MS OS/2 and MS-DOS all in one package.

With Microsoft BASIC Compiler version 6.0, C version 5.1, COBOL version 3.0, FORTRAN version 4.1, Macro Assembler version 5.1, and Pascal version 4.0, you can write programs that break the 640K barrier of DOS. Your MS OS/2 programs can use the entire system memory—up to 16 MB of physical memory and one gigabyte of virtual memory! And they can make direct calls to the operating system. So instead of using a low-level interrupt interface, you can call the operating system just as you would call any other routine.

Microsoft C version 5.1, COBOL version 3.0, FORTRAN version 4.1, Macro Assembler version 5.1, and Pascal version 4.0 can also create Family API (FAPI) programs. FAPI programs run under MS-DOS 3.x or in the MS OS/2 DOS Compatibility Environment in real mode or under MS OS/2 in protected mode. This capability lets you write one version of your program that runs under both MS-DOS and MS OS/2.

Microsoft C version 5.1, COBOL version 3.0, and Macro Assembler version 5.1 also support other major features of MS OS/2, including the ability to create Dynamic Link Libraries and multithreaded programs.

Dynamic Link Libraries (DLLs) are load-time or run-time libraries. Dynamic linking with DLLs takes place after your application is compiled and linked, so it's smaller and loads faster. And DLLs can be shared by multiple applications, so they consume less system memory.



Threads are the fundamental units of execution within an MS OS/2 program. Writing programs with multiple threads allows you to divide an application's functions into separate concurrent operations for more responsiveness. For example, a spreadsheet can recalculate *while* numbers are being entered.

Developing under MS OS/2 also has advantages for you when you program. Multitasking lets you compile and link in the background while you edit another file in the foreground. Large memory support lets you keep all your program modules and development tools loaded and ready to use. And dynamic linking makes your programs smaller and easier to update.

Microsoft languages for MS OS/2 also come with tools that make writing protected-mode programs faster and easier.

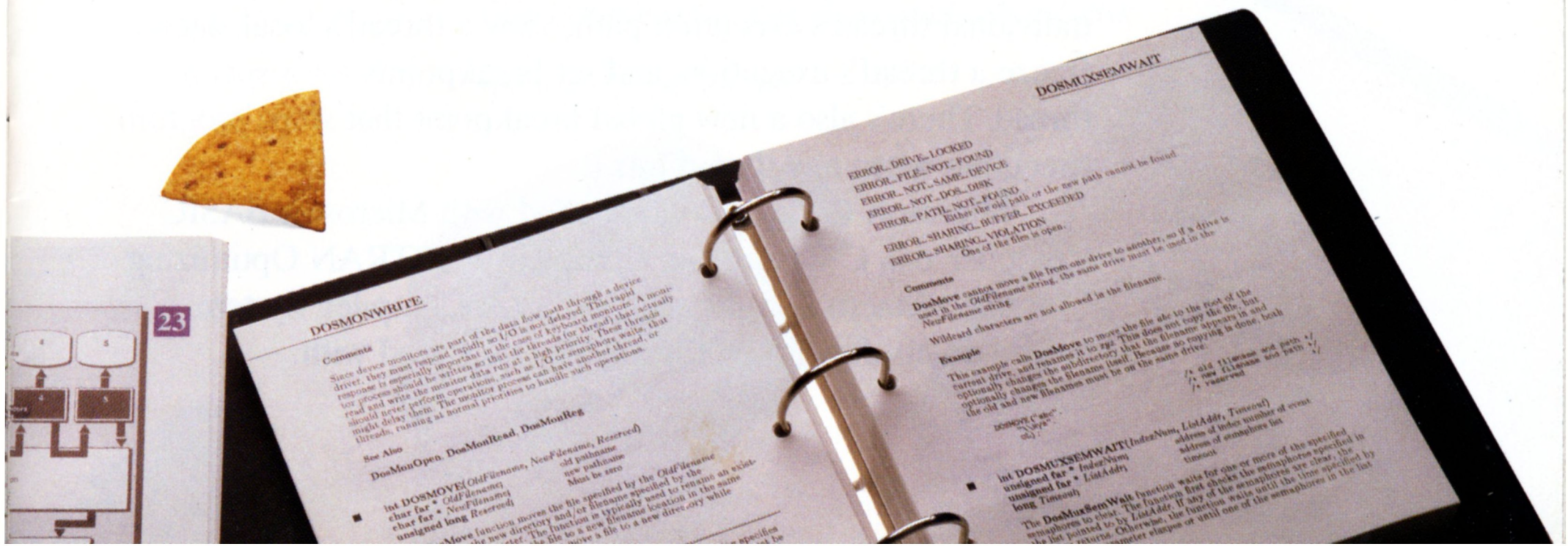
ILINK is the new incremental linker for MS OS/2. After an initial linking with the standard MS-DOS linker, ILINK relinks only the changed modules. An incremental link can be up to 20 times faster than a full link.

Microsoft Editor is the *first* text editor for programmers that lets you take advantage of the power of MS OS/2. But it still runs under MS-DOS as well. Microsoft Editor is a "smart" editor. Not only can you run programs from within it, Microsoft Editor can be set up to generate a compile that reports compilation errors directly back into your source code*—even errors generated from a multiple-module build using MAKE, the program maintenance utility.

Microsoft Editor is programmably extendable. You can write your own editing functions in C and tie them into Microsoft Editor to add any feature you like.

You can write macros for Microsoft Editor on the fly—they don't need to be compiled. And Microsoft Editor can be customized to work like your favorite editor. It even comes with built-in keystroke mapping for Microsoft QuickBASIC, Microsoft QuickC, BRIEF®, Epsilon™, and WordStar.

*Except for COBOL Optimizing Compiler Version 3.0.

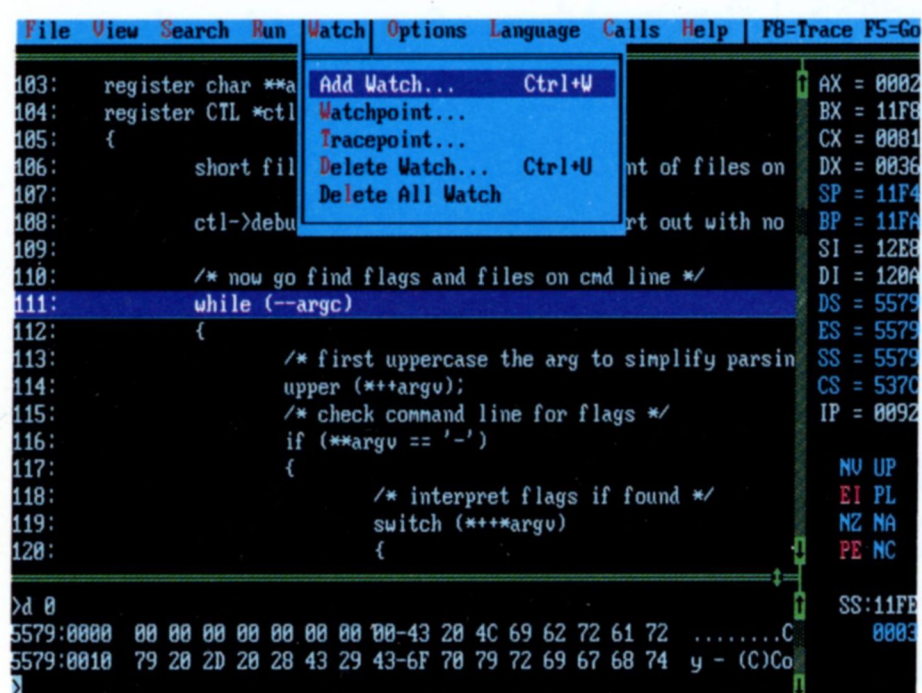


Microsoft CodeView®.

A new experience in debugging—included with many Microsoft languages.

Debugging programs has *never* been this easy. It's Microsoft CodeView, a revolutionary new debugger for programs written in Microsoft QuickBASIC 4.0 and BASIC Compiler 6.0, Microsoft C and QuickC, Microsoft FORTRAN, Microsoft Macro Assembler, and Microsoft Pascal 4.0.

Microsoft CodeView features multiple on-screen windows



Debugging is fast and efficient with Microsoft CodeView. Multiple windows, drop-down menus, and sophisticated features put you in complete control of program execution.

that let you view program execution while you watch variables and register values change. And CodeView displays data structures and interactively follows linked lists and nested structures. Conditional breakpoints stop program execution when a variable or expression reaches a critical value. So a single command can substitute for dozens of traditional breakpoints. With Microsoft CodeView, you can view disassembled code with or without symbols, source and disassembled code intermingled, or

source code only. And drop-down menus mean you don't have to memorize cryptic commands; just use the keyboard or a mouse to activate them.

Microsoft CodeView also supports mixed-language programs, letting you debug in the language you programmed in using your own variable names. You can evaluate language-specific expressions and even call program functions right from the keyboard.

Microsoft CodeView for MS OS/2 provides new capabilities that let you debug protected-mode applications and Dynamic Link Libraries as large as 128 MB. Its support for debugging multithreaded applications includes the ability to trace an individual thread's execution path, view a thread's local stack, freeze a thread's execution, and set breakpoints for a specific thread. There's also a new global breakpoint that stops program execution when any thread hits it.

Microsoft CodeView is included with Microsoft BASIC Compiler 6.0, C Optimizing Compiler, FORTRAN Optimizing Compiler, Macro Assembler, and Pascal Compiler. System requirements depend on the language it's used with.

Microsoft C Optimizing Compiler.

First with the pros.

It's the ultimate C development environment for the IBM Personal Computer, producing code with execution speed second to none. It's Microsoft C Optimizing Compiler.

Microsoft C includes optimization features to reduce the number of instructions in your programs and streamline your code for the targeted hardware. Take advantage of inline code generation of functions, loop invariant expression removal, automatic register allocation of variables within loops, elimination of common sub-expressions, and improved constant folding and value propagation.

The compiler also supports fast, accurate program development. Use Microsoft QuickC's integrated editor, compiler, and debugger (included in the package) for fast prototyping of your C programs—not to mention compilation at 10,000 lines per minute.* The revolutionary Microsoft CodeView window-oriented debugger is included to provide powerful source-level debugging. And there's even documentation specifically designed to help you write the fastest code possible.

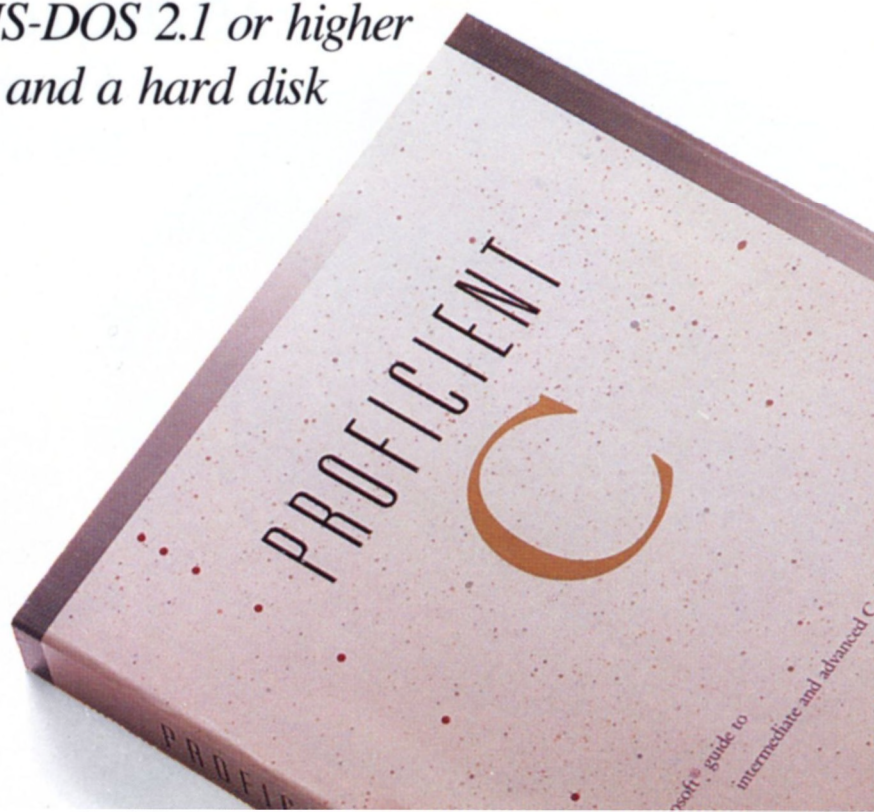
Microsoft C Optimizing Compiler also features:

- *Fast linking (twice as fast as version 4.0)*
- *Enhanced support of the proposed ANSI C standard*
- *Support for Intel 8087/80287/80387 math coprocessors*
- *Interlanguage calling to other Microsoft languages, including QuickBASIC, BASIC Compiler 6.0, FORTRAN, Macro Assembler, and Pascal.*
- *Run-time library source code available separately*

System requirements

- *448K available user memory (512K recommended)*
- *MS OS/2 1.0 or higher or MS-DOS 2.1 or higher*
- *One double-sided disk drive and a hard disk*

*On an IBM PC/AT at 8 MHz.



Microsoft FORTRAN Optimizing Compiler.

Fast and Full FORTRAN.

Mainframe power on a personal computer—that's Microsoft FORTRAN Optimizing Compiler.

Microsoft FORTRAN generates extremely compact code for programs that run faster than you thought possible on a personal computer. And you can transport your programs to other systems easily, because this is a complete implementation of the ANSI FORTRAN 77 standard. Plus we've added the largest set of IBM VS and DEC® VAX® extensions available for personal computers.

Microsoft FORTRAN Optimizing Compiler has been certified as Full and error-free by the National Bureau of Standards (formerly the General Services Administration)—your guarantee of the highest level of reliability you can get in a FORTRAN compiler. And Microsoft FORTRAN comes with the powerful Microsoft CodeView debugger for unprecedented control over your debugging.

Microsoft FORTRAN Optimizing Compiler also features:

- *Math coprocessor and emulator support*
- *Extensive plain-English diagnostic error messages*
- *Direct interlanguage calling to other Microsoft languages, including QuickBASIC, BASIC Compiler 6.0, Quick C, C, Macro Assembler, and Pascal.*
- *Medium, large, huge, and mixed-memory modules*
- *Huge arrays and common blocks greater than 64K*
- *Support for 8087/80287 math coprocessors*
- *Networking support with file and record locking*

System requirements

- *320K available user memory (512K recommended)*
- *MS OS/2 1.0 or higher or MS-DOS 2.1 or higher*
 - *Two double-sided disk drives or one double-sided disk drive and a hard disk*



Microsoft Macro Assembler.

Maximum power for microcomputers.

Microsoft Macro Assembler is an indispensable tool for writing assembly subroutines for Microsoft QuickBASIC, Microsoft BASIC Compiler 6.0, Microsoft QuickC, Microsoft C, Microsoft FORTRAN, and Microsoft Pascal. With Microsoft Macro Assembler, you can optimize frequently used subroutines as well as those not easily implemented in high-level languages. Built-in directives make it easy to develop and maintain assembler code for the Intel 80386 and related microprocessors under both MS OS/2 and MS-DOS.

The *Mixed-Language Programming Guide* included in the package shows you how—it's just part of the comprehensive documentation that makes Microsoft Macro Assembler easy to learn.

Debugging your assembly-language programs is faster and easier than ever with Microsoft CodeView, and it's right in your Microsoft Macro Assembler package. Use CodeView's power to see your source and disassembled code simultaneously, including labels, constants, and comments. Debug mixed-language programs, programs with overlays, and large programs that use the Expanded Memory Specification (EMS).

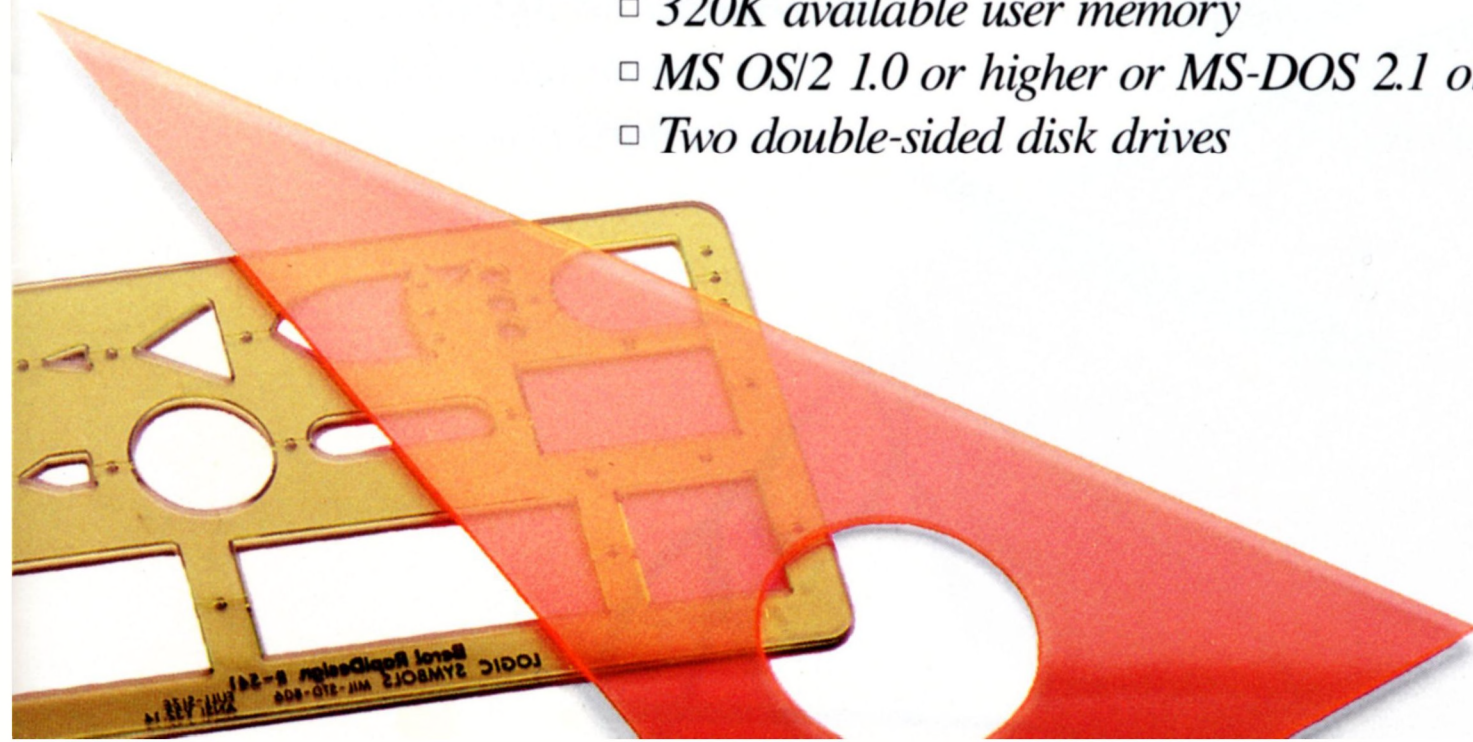
And Microsoft Macro Assembler is the most powerful development tool available for the Intel 8086/80286/80386 processors and 8087/80287/80387 math coprocessors.

Microsoft Macro Assembler also features:

- *Access to all available memory on your personal computer for assembling complex programs*
- *Simplified segment directives*
- *Default IEEE floating-point format (binary format optional)*

System requirements

- *320K available user memory*
- *MS OS/2 1.0 or higher or MS-DOS 2.1 or higher*
- *Two double-sided disk drives*



Microsoft BASIC Compiler.

The total professional BASIC for MS OS/2.

If programming in BASIC is your business, Microsoft BASIC Compiler is for you. It's the first BASIC compiler for MS OS/2 that gives you everything a professional programmer needs. You get the most advanced BASIC language implementation (including support for structured programming), Microsoft CodeView for powerful source-level debugging, and Microsoft QuickBASIC 4.0 for fast program development.

Microsoft BASIC Compiler also features a flexible run-time library. You control which portions of the run-time library are linked to your program, and if you use the separate run-time module, you can customize it with your own common routines to save disk space.

Microsoft BASIC Compiler also features:

- *Alternate math library for added speed on systems without a math coprocessor (8087/80287 coprocessor support and a floating-point emulator are also included)*
- *User-defined events for test instrumentation and industrial control applications*
- *Enhanced language features, including MS OS/2 language additions, user-defined types, recursion, and support for huge arrays*
 - *Interlanguage calling to other Microsoft languages, including Microsoft QuickBASIC, QuickC, C, FORTRAN, Macro Assembler, and Pascal*

System requirements

- *320K available user memory*
- *MS OS/2 1.0 or higher or MS-DOS 2.1 or higher (IBM Personal Computer or compatible required for MS-DOS)*
 - *One double-sided disk drive and a hard disk*



Microsoft Pascal Compiler.

When you've outgrown the others.

If you want a Pascal that can push your personal computer to its limits, move up to the Microsoft Pascal Compiler. It's based on the proposed ISO and ANSI Level 0 standards to ensure language compatibility—then we've added extensions to better support the development of large, powerful programs.

Microsoft Pascal's symbol table can be as large as available memory, letting you write larger Pascal programs for the MS OS/2 and MS-DOS operating systems and the Microsoft Windows Presentation Manager. You can easily develop your programs in modules for fast coding and compiling, then link them together to create large applications. And your code can be combined with other modules developed in other Microsoft languages, including QuickBASIC, BASIC Compiler 6.0, QuickC, C, FORTRAN, and Macro Assembler (some limitations apply), letting you choose the best language for each task.

Microsoft Pascal Compiler includes our Microsoft CodeView window-oriented debugger for fast, efficient program development.

Microsoft Pascal Compiler also features:

- *8087/80287 math coprocessor and emulation support*
- *Source-code XENIX compatibility*
- *Multiuser file sharing with file and record locking*

System requirements

- *320K available user memory (512K recommended)*
- *MS OS/2 1.0 or higher or MS-DOS 2.1 or higher*
- *Two double-sided disk drives or one double-sided disk drive and a hard disk*

Microsoft COBOL Optimizing Compiler.

The complete COBOL solution.

When you mean business, you mean COBOL. And when you mean COBOL, you want the complete compiler—Microsoft COBOL Optimizing Compiler. Not only can you write standard MS-DOS applications, you can write protected-mode programs for MS OS/2 that break the 640K barrier of DOS. Microsoft COBOL is ANSI 85 COBOL certified at the High level by the

National Bureau of Standards (NBS). And it's a native-code compiler, so you're assured of the fastest execution possible.

Microsoft COBOL Optimizing Compiler is compatible with seven COBOL dialects, including IBM VS COBOL II™, IBM OS/VS™ COBOL, and RM/COBOL®. So you can develop applications on your personal computer and port them to a mainframe. Or port mainframe programs to your personal computer.

You can also take advantage of structured language programming enhancements such as the in-line PERFORM, EVALUATE, and INITIALIZE statements, as well as scope delimiters, negated conditions, global variables, reference modifications, and nested programs. The ANIMATOR™ source-code debugger supports backtracking, breakpoint DO statements, and periodic breakpoints. And the HUGE memory model allows data items to be greater than 64K.

Microsoft COBOL Optimizing Compiler also features:

- *Microsoft Editor, the first programmer's editor for both MS OS/2 and MS-DOS*
- *ILINK, the incremental linker for MS OS/2 that performs partial links up to 20 times faster than a full link—only changed modules are relinked*
- *Support for the development of large mainframe COBOL applications on a personal computer*

System Requirements

- *384K available user memory (512K recommended)*
- *MS OS/2 1.0 or higher or MS-DOS 3.0 or higher*
- *One double-sided disk drive and a hard disk*

```
299 process-input-data.
300   open input month-result-file
301   if first-time move zero to month-count, tax-paid end-if
302   perform with test after varying month-index from 1 by 1
303     until month-index = 12
304     read month-result-file
305     at end perform no-results
306   end-read
307   add month-profit to refunds giving gross-profit
308   move gross-profit to total-profit (month:6)
309   evaluate month
310     when January
311       perform year-begin
312       perform first-half
313     when February through June
314       perform first-half
315     when other
316       perform second-half
317   end-evaluate
318 end-perform
319 perform
```

Animator-A/CMD... Ins Caps Num Scroll
F1 Help F2 View F3 Edit F4 Exchange F5 Show F6 Hide F9 F10 Esc
Stop Go Zoom Next I Prev R B E Q F L T D 0 9

The ANIMATOR source-code debugger lets you trace forward or backward, execute COBOL statements at any point in the program, and activate breakpoints on a specified iteration of a statement's execution.

Microsoft muMATH™.

Mainframe math on your personal computer.

Going beyond the math capabilities found in traditional programming languages, the Microsoft muMATH Symbolic Mathematics Package efficiently and accurately performs both symbolic and numerical manipulations—it can even evaluate and simplify expressions containing variables that have not been assigned numeric values. Built-in trigonometry, calculus, matrix operations, vector, and other functions tackle the toughest algebraic and analytic problems. The package also includes muSIMP, the language in which muMATH was written, so you can add customized features. Microsoft muMATH is the symbolic mathematics system for your personal computer.

System requirements

- 128K memory
- MS-DOS 2.0 or higher
- One double-sided disk drive



Microsoft Operating System/2 Programmer's Toolkit.

Bring the full power of MS OS/2 to your applications.

Create the next generation of personal computer applications with the MS OS/2 Programmer's Toolkit. It provides you with all the tools you need to develop applications for MS OS/2. When you add a Microsoft language with OS/2 system support such as BASIC Compiler 6.0, C 5.1, FORTRAN 4.1, Macro Assembler 5.1, or Pascal 4.0, you have a complete OS/2 system development environment.

Under MS OS/2, you can write programs that break the 640K barrier of DOS and have multiple threads of execution. Support for protected mode also means that MS OS/2 can run multiple applications simultaneously—a great increase in productivity for you as well as end users.

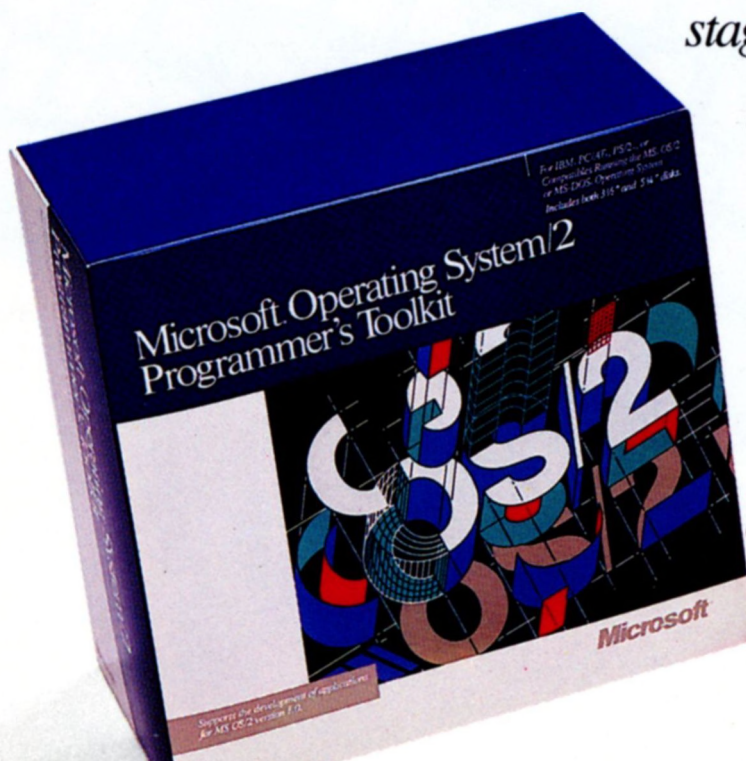
MS OS/2 features a new call-based applications program interface (API) that lets you call the operating system directly instead of with interrupts. And your applications can share Dynamic Link Libraries, saving data space and memory.

The Microsoft Operating System/2 Programmer's Toolkit also features:

- *BIND and API.LIB, which let you write Family API applications that run under both MS OS/2 and MS-DOS*
- *IMPLIB for creating Dynamic Link Libraries*
- *Sample MS OS/2 programs with executables and source code*
- *Detailed, easy-to-use documentation*
 - *MS OS/2 Programmer's Reference defines all MS OS/2 function calls, structures, and file formats (also provided as on-line, context-sensitive help)*
 - *MS OS/2 Programming Tools explains how to use the utilities and tools to create MS OS/2 programs*
 - *MS OS/2 Programmer's Learning Guide leads you through the stages of building an MS OS/2 application*

System Requirements

- *2 MB memory*
- *MS OS/2 1.0 or higher or IBM OS/2 1.0 or higher*
- *One hard disk and one high-density (1.2 MB) 5¼" disk drive or one high-density (1.44 MB) 3½" disk drive*



Microsoft Windows Software Development Kit.

A full set of tools for building dynamic graphics-based applications.

The Microsoft Windows Software Development Kit gives you the tools you need to develop graphics-based applications for Microsoft Windows/286 and Microsoft Windows/386. Combine the software development kit with a Microsoft programming language such as C Optimizing Compiler, Macro Assembler (MASM), or Pascal Compiler and a Microsoft Windows retail package for a complete development environment.

The Microsoft Windows Software Development Kit comes with a full set of sophisticated development tools, including the Dialog Editor, Icon Editor, Font Editor, resource compiler, symbolic debugger, and MAKE—the program maintenance utility. And the enhanced application programming interface (API) makes it easier to write Windows applications.

Microsoft Windows Software Development Kit also features:

- *User interface for your applications that complies with IBM's Systems Applications Architecture (SAA) and is visually consistent with the OS/2 systems' user interface*
- *Support for the Intel 80287 math coprocessor*
- *Description of Dynamic Data Exchange (DDE), which lets your applications communicate with other Windows applications*
- *Definition of Tagged Image File Format (TIFF), which standardizes the storing and transferring of scanned and graphic images*
- *Revised Programmer's Reference for easier, faster information access*
- *Programmer's Learning Guide, which leads you through the stages of building a Windows application (source code is included)*

System Requirements

- *640K memory*
- *DOS 3.0 or higher*
 - *One 1.2 MB disk drive*
 - *One hard disk with 2 MB available storage space*
 - *Graphics adapter card (Hercules® Graphics Card, IBM CGA, IBM EGA, IBM VGA, or compatible)*
 - *One of the following Microsoft languages:*
 - *Microsoft C Optimizing Compiler version 4.0 or higher*
 - *Microsoft Macro Assembler version 4.0 or higher*
 - *Microsoft Pascal Compiler version 4.0 or higher*
 - *Microsoft Mouse or compatible (required for some tools)*
 - *Microsoft Windows/286 version 2.1 or higher or Microsoft Windows/386 version 2.1 or higher*



Microsoft Project.

Keep your projects on schedule.

Microsoft Project gives you an accurate overview of all sizes of programming projects, helping you keep them on time and within budget. It lets you plan activities, assign resources, monitor schedules and costs, and report project information quickly.

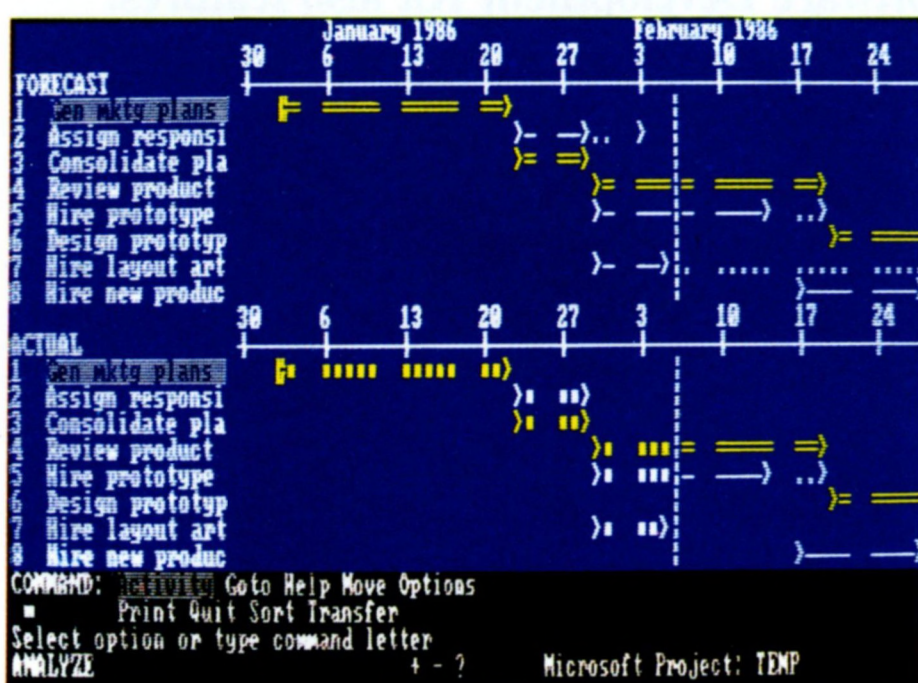
With Microsoft Project, you can choose the best chart format for your needs—PERT or Gantt—and compare planned and actual schedules to track your progress. A flexible report generator lets you customize report formats and content to communicate schedule status effectively. And the program can output to a plotter as well as a printer for presentation-quality charts.

Microsoft Project also offers informative resource histograms for comparing resource assignments and loading, as well as individual resource calendars to factor in vacations and other

time off. Its resource-leveling feature automatically resolves resource scheduling conflicts. And Microsoft Project comes with a complete computer-based tutorial so it's as easy to learn as it is to use.

System requirements

- 256K memory
- MS OS/2 1.0 or higher or MS-DOS 2.1 or higher
- Two double-sided disk drives or one double-sided disk drive and a hard disk



Microsoft Project helps you monitor your projects and track your progress by letting you compare your actual schedule with your planned schedule.



Microsoft languages for the XENIX operating system.

Portable. Powerful. Practical.

Microsoft offers powerful implementations of our most popular languages for the XENIX 286 operating system.

Microsoft BASIC Interpreter.

For the benefits of BASIC programming without compiling, turn to Microsoft BASIC Interpreter. It offers on-line debugging, allows access to the XENIX file system, and features a high degree of compatibility with the Microsoft BASIC Interpreter and GW-BASIC® Interpreter for MS-DOS systems.

Microsoft BASIC Compiler.

Bringing the speed and power of compiled BASIC to the XENIX environment, this product lets you compile programs created with the Microsoft BASIC Interpreter so you can run them up to ten times faster. It's also fully compatible with the MS-DOS version and gives you a wide range of support for XENIX file- and screen-handling functions.

Microsoft FORTRAN Compiler.

This native-code compiler provides a wide range of ANSI FORTRAN 77 features and extensions and allows source code transfer to programs written with Microsoft FORTRAN for the MS-DOS operating system. It supports the use of math coprocessors and lets you develop applications from convenient, easily linked modules.

Microsoft Pascal Compiler.

Put the full power of your personal computer to work with Microsoft Pascal Compiler. Create large programs easily by linking separate modules and using overlays. Incorporate sub-routines from other languages. Use data up to the limit of available memory. And the source code is compatible with Microsoft Pascal Compiler for the MS-DOS operating system.

Microsoft Press

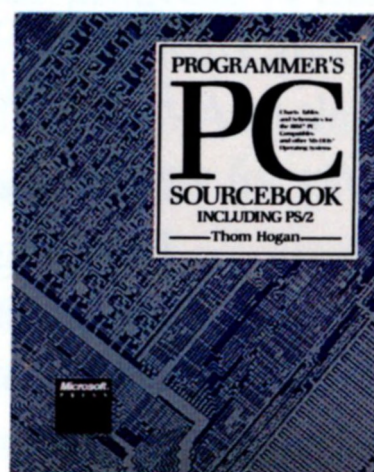
The last word on languages.

To make your programming even more efficient, Microsoft Press offers books on individual languages as well as on the basics of programming. Check out the full selection at your local book-seller or software store, or call (800) 638-3030 to place your credit card order.



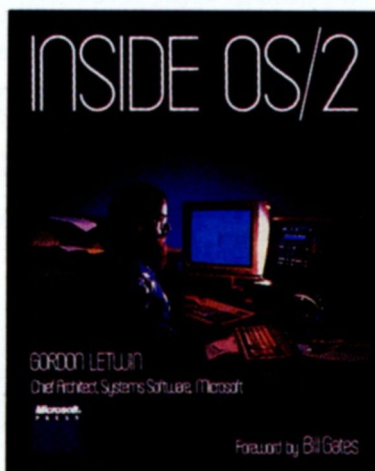
Proficient C
Augie Hansen

Aimed at intermediate to advanced programmers, *Proficient C* focuses on techniques and programming tools that can be used to develop fast, powerful C applications. Emphasis is placed on creating reusable modules that have a wide range of uses. Source code and executable-program companion disks are available.



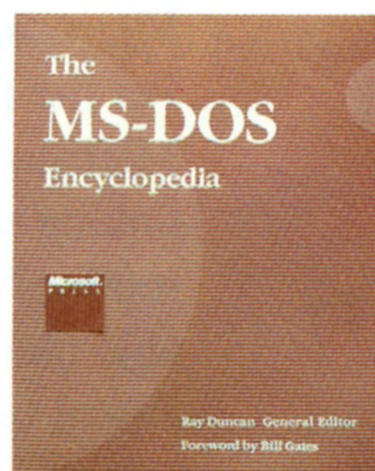
The Programmer's Sourcebook
Thom Hogan

The Programmer's Sourcebook is a convenient, easy-to-use reference designed to save every programmer hours of frustration. Dozens of charts and tables provide valuable information on the MS-DOS and MS OS/2 operating systems and on all IBM personal computers, including the new Personal System/2[®] machines.



Inside Microsoft OS/2
Gordon Letwin

Written by Gordon Letwin, Microsoft's chief architect for MS OS/2, *Inside Microsoft OS/2* provides a true behind-the-scenes look at how Microsoft is merging its vision of an automated office system with the realities of software development. You can't get a more inside view of the MS OS/2 operating system than this.



MS-DOS Encyclopedia
(Versions 1.0 through 3.2)

Foreword by William H. Gates III, Chairman, Microsoft Corporation. The *MS-DOS Encyclopedia* contains detailed and accurate information on the MS-DOS operating system, the system used on more than 10 million computers worldwide. The encyclopedia was written in close association with the Microsoft MS-DOS development team and includes contributions from noteworthy industry authorities.

Which Microsoft language do you want to speak?

Find out more about the Microsoft languages that interest you. Just mail in this coupon, and we'll send you detailed technical data sheets on the Microsoft languages of your choice. You can also request information on Microsoft Project, our project-scheduling program.

- ☐ Microsoft® QuickBASIC (MS-DOS only)
- ☐ Microsoft BASIC Compiler (MS-DOS® and MS® OS/2)
- ☐ Microsoft BASIC Compiler (XENIX®)
- ☐ Microsoft BASIC Interpreter (XENIX only)
- ☐ Microsoft QuickC™ (MS-DOS only)
- ☐ Microsoft C (MS-DOS and MS OS/2)
- ☐ Microsoft COBOL (MS-DOS and MS OS/2)
- ☐ Microsoft FORTRAN (MS-DOS and MS OS/2)
- ☐ Microsoft FORTRAN (XENIX)
- ☐ Microsoft Macro Assembler (MS-DOS and MS OS/2)
- ☐ Microsoft muMATH™
- ☐ Microsoft Pascal (MS-DOS and MS OS/2)
- ☐ Microsoft Pascal (XENIX)
- ☐ Microsoft Project

To order, send this form to Microsoft Corporation/
Attention: Consumer Response/16011 NE 36th Way/
Box 97017/Redmond, WA 98073-9717. Please print.

Name _____

Company _____

Address _____

City _____

State _____

ZIP _____

Phone (_____) _____



Microsoft Corporation
16011 NE 36th Way
Box 97017
Redmond, WA 98073-9717
USA

Microsoft Pty Ltd
Sydney AUSTRALIA

Microsoft Canada Inc
Toronto CANADA

Microsoft Ltd
Reading ENGLAND

Microsoft SARL
Paris FRANCE

Microsoft BV
Hoofddorp HOLLAND

Microsoft Ireland
Dublin IRELAND

Microsoft SpA
Milan ITALY

Microsoft KK
Tokyo JAPAN

Microsoft SA
Seoul KOREA

Microsoft Mexico
Mexico DF MEXICO

Microsoft AB
Spanga SWEDEN

Microsoft GmbH
Munich WEST GERMANY

© Copyright 1988, Microsoft Corporation. All rights reserved. Printed in USA.

Please note that the availability and specifications of products described here are subject to change without notice.

Microsoft, the Microsoft logo, MS, MS-DOS, XENIX, GW-BASIC, and CodeView are registered trademarks and QuickC is a trademark of Microsoft Corporation.

ANIMATOR is a trademark of Micro Focus Limited. BRIEF is a registered trademark of UnderWare, Inc. DEC and VAX are registered trademarks of Digital Equipment Corporation. Epsilon is a trademark of Lugaru Software Ltd. Hercules is a registered trademark of Hercules Computing Technology. IBM, Personal System/2, and PC/AT are registered trademarks and OS/VS and VS COBOL II are trademarks of International Business Machines Corporation. Intel is a registered trademark of Intel Corporation. MuMATH is a trademark of Soft Warehouse, Inc. RM/COBOL is a registered trademark of Ryan McFarland, Inc. WordStar is a registered trademark of MicroPro International.

Microsoft COBOL Optimizing Compiler version 3.0 is based on COBOL/2 from Micro Focus.

0588 Part No. 02331